Beyond Single Object: Learning 3D Relations with Large Language Models – Supplementary Material –

Kohsuke Ide[1,2] Ryousuke Yamada[1,3] Yue Qiu[1] Xianzheng Ma[4]
Yoshihiro Fukuhara[1] Hirokatsu Kataoka[1,4] Yutaka Satoh[1,2]
[1]AIST [2]University of Tsukuba
[3]University of Technology Nuremberg [4]University of Oxford

1. Implementation Details

Model Checkpoints

We initialized our system using publicly available pre-trained models from the PointLLM [21] framework:
Language Model: Vicuna-7B-v1.5 (lmsys/vicuna-7b-v1.5) [6]
Point Cloud Encoder: Point-BERT [23], pre-trained via ULIP2 [22]
CLIP used for classification, grouping: OpenCLIP ViT-L/14 backbone [18]

1.1. Training Hyperparameters

We detail the hyperparameters used for our two-stage training strategy in Table 1. All models were trained using the AdamW optimizer with a cosine learning rate scheduler and a warmup ratio of 0.03.

Table 1. **Training Hyperparameters.**

Parameter	Phase 1 Feature Alignment	Phase 2 Holistic Task-Mixture
Batch Size	16	14
Learning Rate	2e-3	2e-5
Weight Decay	0.0	0.0
Number of Epochs	3	3
LR Scheduler	Cosine w/ Warmup	Cosine w/ Warmup
Warmup Ratio	0.03	0.03
Trained Params	Projector	Projector, PIT block, LLM
Frozen Params	Point Encoder PIT block, LLM	Point Encoder

Hardware

Training was conducted on 8 NVIDIA H200 GPUs (140GB VRAM).
Training Time. The training process was completed in two phases with the following durations: 70 minutes for feature alignment (Phase 1), 12 hours for Training on holistic mixture of datasets (MO3D, Shape Mating, Change Captioning).

2. Dataset Generation Details

2.1. MO3D Dataset

Data-Driven Category Definition. To ensure our benchmark covers diverse aspects of 3D objects, we established six core categories through a two-stage process. First, we prompted *Qwen2-72B-Instruct* [2] to identify common attribute types from 70k samples of Objaverse-Cap3D [7, 15] captions. Then, we manually grouped these outputs into six semantically distinct and comprehensive categories, followed by manual curation. These categories, detailed in Table 4, form the foundation for our balanced generation pipeline.

Table 2. **Overall Statistics of the MO3D Dataset.** The dataset maintains a perfectly balanced distribution across the three core tasks and a near-even split between positive and negative object groupings.

Total Samples	Task Types			Group Types	
	Positional	Comparative	Holistic	Positive	Negative
69,996 (100%)	23,332 (33.3%)	23,332 (33.3%)	23,332 (33.3%)	34,926 (49.9%)	35,070 (50.1%)

Group Formation Strategy. The foundation of each query is a meaningful group of objects. We formulated these groups based on the semantic similarity of captions from the Objaverse-Cap3D corpus. To quantify similarity efficiently at scale, we computed embeddings using a pre-trained CLIP text encoder (ViT-L/14) [18] and leveraged Faiss [12] to pre-compute and cache the top-50 nearest neighbors for every object.

At generation time, we select an anchor object and $n-1$ companions. We specifically focus on group sizes of $n \in \{2, 3\}$. The inclusion of 3-object groups is a deliberate design choice: a group of three is the minimal configuration required to introduce complex relational concepts, such

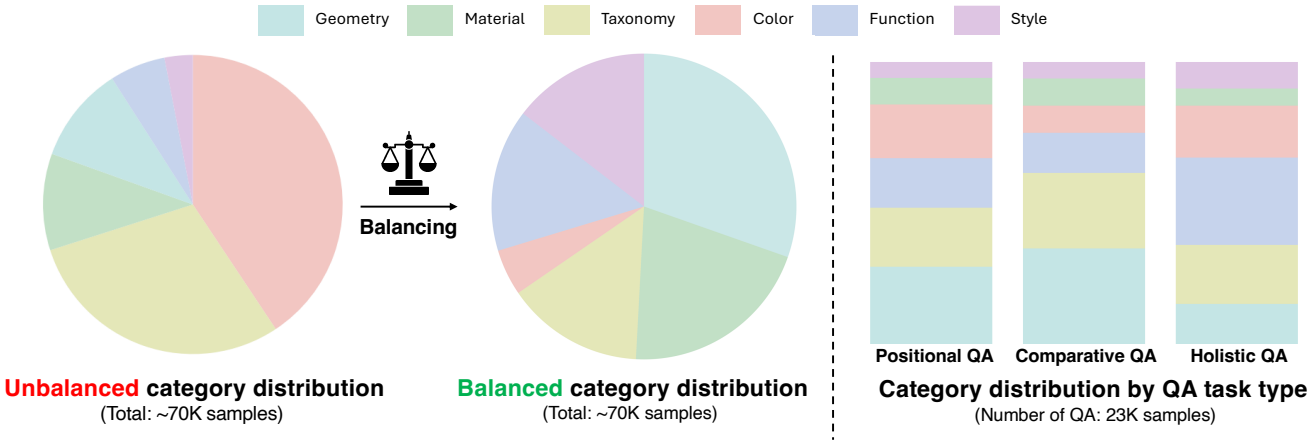


Figure 1. **Category Distribution and Weighted Balancing.** We transform the naturally occurring, unbalanced attribute distribution (left) into a curated target distribution (middle). This process explicitly prioritizes geometric and structural categories (*e.g.*, Geometry, Material) while suppressing superficial visual cues (*e.g.*, Color) to enforce 3D understanding. The stacked bars (right) verify that this balanced distribution is consistently applied across all three task types: Positional, Comparative, and Holistic QA.

Table 3. **Distribution of Question Categories.** We compare our design target weights with the actual distribution in the final dataset. The resulting distribution closely matches our goal of prioritizing geometric and structural comparison.

Category	Target Weight	Actual Count (%)
Geometry / Structure	30%	21,264 (30.4%)
Material	20%	14,316 (20.5%)
Function	15%	10,578 (15.1%)
Taxonomy	15%	10,200 (14.6%)
Style / Aesthetics	15%	10,206 (14.6%)
Color	5%	3,432 (4.9%)

as identifying a semantic "outlier" or finding a "majority" property, which are impossible with only pairwise comparisons. To ensure a balanced distribution of comparison scenarios, we employ a Dual Sampling strategy with specific thresholds:

- Positive Sampling: Selects companion objects from the anchor's top-50 semantically similar neighbors. This results in groups with high conceptual overlap, suitable for fine-grained comparison (e.g., distinguishing between two different chairs).
- Negative Sampling: Selects companions from the pool of items that do not appear in the anchor's top-50 neighbors. This produces groups with low conceptual overlap for broader distinctions.

Finally, the sequence of selected objects is randomly shuffled to ensure the task is order-invariant and to prevent the model from learning positional biases.

Guided Question Generation. We use GPT-4 to generate QA pairs, inputting both the Cap3D captions and multi-view renderings. To prevent hallucinations and ensure quality, we enforce a strict Prompt Hierarchy:

- Grounding Constraint (Highest Priority): All information must be strictly visually grounded in the provided inputs. Invented details are prohibited.
- Task-Specific Constraints : Each task type (e.g., geometry) enforces specific keywords (e.g., use "shape" instead of ambiguous "feature").
- Category Coverage: We use weighted balancing to target specific categories, prioritizing geometric understanding over simpler visual cues as shown in Figure 1. If a category is not applicable (e.g., no color information), the model falls back to a valid alternative.

The full prompt used for generation is provided in Figure 5.

Balance Correction. To mitigate linguistic priors (*e.g.*, the tendency to answer "Yes"), we implement a post-hoc balance correction. We analyze the `holistic` subset and identify any imbalance. We then regenerate a subset of samples using a constrained prompt that forces the generation of a "No" question (e.g., asking about a property *not* shared by the group), ensuring a near 50:50 distribution in the final dataset.

Dataset Statistics. The final MO3D dataset consists of approximately 70k multi-object instruction-following examples. As detailed in Table 2, the dataset is perfectly balanced across the three core tasks (Positional, Comparative, Holistic), with each constituting exactly one-third of the data. The object grouping strategies (Positive vs. Negative sampling) are also balanced near 50:50 to ensure diverse comparison scenarios. Furthermore, Table 3 demonstrates that our weighted sampling strategy successfully aligned the generated question categories with our target distribution, prioritizing geometric and structural understanding (30.4%) over simpler attributes like color (4.9%). We further analyze the linguistic complexity in Table 5 and Figure 8.

Table 4. The six core reasoning categories curated from our data-driven analysis. We define each category and provide examples of the raw, LLM-discovered themes that were merged into it.

Core Category	Description and Example Question Focus	Examples of Merged Raw Categories
geometry/structure	The shape, size, number of parts, and structural complexity of an object. *e.g., "Does it have more than four legs?"*	`shape`, `size`, `components`, `part`, `feature`, `features`, `detail`, `base`, `form`
taxonomy	The general class, type, or identity of an object. *e.g., "Is this object a piece of furniture?"*	`objecttype`, `type`, `object_type`, `object`, `theme`, `category`
function	The intended purpose, use, or action associated with an object. *e.g., "What is this object used for?"*	`function`, `action`, `purpose`, `usage`, `activity`
material	The physical substance an object is made of. This is distinct from color. *e.g., "Is the frame made of metal or wood?"*	`material`, `substance`, `composition`
style/aesthetics	The visual style, era, pattern, decoration, or contextual setting of an object. *e.g., "Does this object have a writing on the surface?"*	`style`, `decoration`, `pattern`, `design`, `era`, `location`, `context`
color	The surface color or hue of an object or its parts. *e.g., "Is the main color of the object brown?"*	`color`, `hue`, `finish`

Data Reliability and Intrinsic Ambiguity. To rigorously validate the reliability of the MO3D dataset, we conducted a human audit on 500 randomly sampled QA pairs from the test set. Three independent annotators evaluated each pair, achieving a strong unanimous agreement rate of 81.0%. Through this audit, we found that the remaining disagreements did not stem from incorrect ground truths, but rather from the *intrinsic ambiguity* of 3D object interpretation. As illustrated in Figure 2, subjective perceptions of highly abstract shapes or partial occlusions naturally lead to divergent valid interpretations among humans. This highlights the inherent complexity of 3D relational tasks compared to standard 2D QA. Crucially, when evaluated on this high-quality unanimous subset, our model achieves an average accuracy of 59.2% across the MO3D tasks, closely aligning with our 57.3% average on the full test set. This verifies that intrinsic ambiguities do not artificially distort the evaluation metrics.

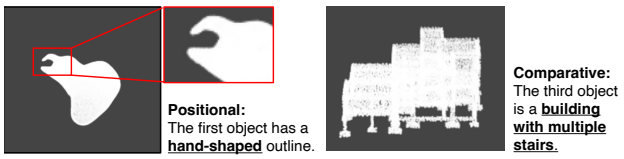


Figure 2. **Intrinsic Ambiguity.** Disagreements in human verification often arise from subjective perception (*e.g.* abstract shapes or occlusions)

Table 5. **Linguistic Complexity Statistics.** We report the average and maximum length in words for instructions and responses. Shape Mating features long responses due to the requirement of geometric rationales.

Dataset	Task Type	Instruction		Response	
		Mean	Max	Mean	Max
MO3D	Positional	10.8	21	10.0	25
	Comparative	13.3	30	18.1	33
	Holistic	9.9	18	12.8	31
Shape Mating	Selection	31.8	34	73.5	130
Change Cap.	Verify	64.3	191	25.4	83
	Delta	16.8	18	25.6	96

2.2. Mini-App A: Shape Mating Details

Data Construction and Sampling. We source base 3D meshes from the Thingi10K dataset [24] and generate mating pairs using the `cut shell` operation from Neural Shape Mating [5]. We utilize five distinct cut geometries: *Planar*, *Sine*, *Square*, *Pulse*, and *Parabolic*. Crucially, we selected the `cut shell` operation to ensure domain consistency with our point cloud encoder, Point-BERT [23]. Since Point-BERT is pre-trained solely on object surface points rather than interior cross-sections, the introduction

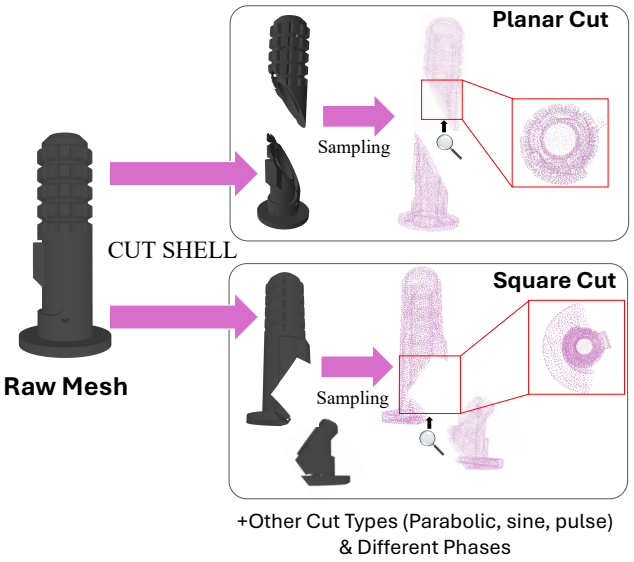


Figure 3. **Point cloud generation pipeline for Shape Mating.** Starting from a raw mesh (Left), we apply the `cut shell` operation to split the object into two complementary halves (Middle). Unlike solid cuts, this operation preserves the hollow, surface-only structure of the object. Finally, each part is uniformly sampled into 8,192 points (Right) to serve as the input for our model.

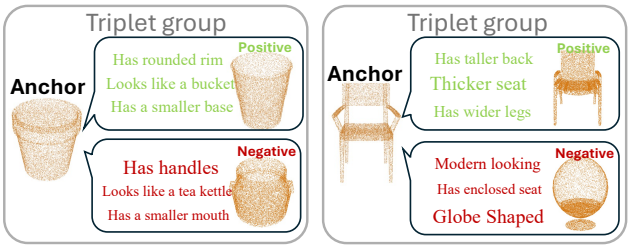


Figure 4. **Hard Negative Sampling for Change Captioning.** We construct contrastive triplets where the *Negative* candidate (red captions) is not random, but a "hard negative" sharing the same source *Anchor* as the *Positive* target (green captions). This Negative sample is crucial for the Verification task to generate challenging "No" instances

of artificial flat cut-faces inherent to solid cutting methods would result in a domain gap. Consequently, we utilize the `cut shell` method to preserve the surface-shell characteristic, thereby aligning the input distribution with the pretraining regime of the encoder. Finally, we perform uniform random sampling on the mesh surface to generate dense point clouds of 8,192 points per part (Figure 3), capturing the fine-grained geometric details required for the mating task.

To strictly enforce the 4-choice classification task, we employ a targeted sampling logic for scene composition. For *1-Mate (Positive)* scenarios, we randomly select a valid ground-truth pair (Part A and Part B from the same instance) and sample a third "decoy" part. This decoy is carefully selected to be non-mating due to specific reasons, such as originating from a different object, a different cut type, or a different cut position (Phase Mismatch), and its position is randomized. For *0-Mate (Negative)* scenarios, we sample three parts such that no combination forms a valid pair, acting as a hard negative that forces the model to verify all possible connections.

Question and Rationale Formulation. We construct the QA pairs using a two-step process to ensure both linguistic diversity and geometric grounding. First, for the question component, we employ a set of 15 distinct templates (detailed in Table 12) to ensure consistent task formulation while providing linguistic variety. These templates explicitly list the four options and mandate a reasoning-based response. Second, for the rationale component, we train the model to explain why a pair does not mate by automatically assigning structured error tags to non-mating pairs. We then use *GPT-4o-mini* to paraphrase these tags into natural language justifications. The full prompt used for this paraphrasing is provided in Figure 6. The error types are defined as follows:

- *cut mismatch* indicates that the two parts possess disparate cut interfaces, such as a planar surface versus a sinusoidal one.
- *object mismatch* signifies that although the parts share the same cut type, they originate from distinct source objects and thus do not align globally.
- *phase mismatch* occurs when parts share the same object and cut geometry but are derived from different cut instances or positions, preventing an exact fit.
- *same side* denotes topological incompatibility, where the selected parts represent the same side of the object, such as two "Part A" components.

Dataset Statistics. We provide a detailed analysis of linguistic complexity in Table 5 and visualize the length distributions in Figure 9. As shown in the table, Shape Mating involves particularly long responses due to the requirement for detailed geometric reasoning.

2.3. Mini-App B: Change Captioning Details

Data Construction and Sampling. We construct this benchmark using the ShapeNet subset of the ShapeTalk dataset [1]. The core unit is a contrastive triplet consisting of an *Anchor*, a *Positive*, and a *Negative* point cloud, paired with an *Instruction*. We employ a strict *Hard Negative Sampling* strategy to ensure difficulty, as illustrated in Figure 4. For a given Anchor-Positive pair (e.g., "thinner back"), we prioritize sampling a Negative shape that shares the same Anchor but corresponds to a different edit instruction (e.g., "thicker seat"). This forces the model to ground the specific semantic details of the instruction, rather than

relying on coarse object recognition. If no such hard negative exists, we fall back to a random object from the same semantic class.

Question and Rationale Formulation. We transform the raw triplet data into model inputs using standardized templates. As detailed in Table 13, we employ distinct template sets for each task to ensure consistent definition while introducing phrasing variety. To ensure high-quality linguistic output, we employ *GPT-4o-mini* for both tasks. For the Verification task, we explicitly randomize the input order of the Anchor and Candidate point clouds to prevent the model from memorizing positional cues (e.g., assuming the answer is always the second object). We used *GPT-4o-mini* to construct a natural language rationale based on the original instruction associated with the Positive/Negative object. For the Delta Captioning task, we similarly use *GPT-4o-mini* to paraphrase and consolidate multiple raw utterances into a single, fluent description.

Dataset Statistics. We report the linguistic complexity in Table 5. Notably, the Verification task has the longest average instruction length of 64.3 words because the input explicitly includes the full list of geometric requirements (from the instruction) that the model must check.

2.4. Data Splitting

Across all our benchmarks (MO3D dataset, Shape Mating, and Change Captioning), we employ a strict leakage-free splitting strategy to ensure rigorous evaluation, as simple random splitting is insufficient when 3D assets share underlying geometries or appear in multiple grouping scenarios. To prevent data leakage, we utilize a graph-based approach where every unique 3D asset is represented as a node, and edges are drawn between nodes if they appear together in the same sample (*e.g.*, within a triplet) or share the same source object. We then compute the connected components of this graph and atomically assign entire components to a single split (Train or Test). This methodology guarantees that no object instance, nor any of its co-occurring or geometrically related variants, ever leaks across splits, ensuring that our evaluation measures true generalization rather than memorization.

3. Additional Ablation Studies

3.1. Baselines & Input Format Fairness

To ensure a fair comparison across fundamentally different architectures, we carefully designed modality-specific input formatting strategies:

Object-Centric 3D-LLMs: Models like PointLLM and ShapeLLM natively accept a single point cloud. For our multi-object tasks, we implement a *concatenation with separation* strategy. Each object's point cloud is individually normalized into a unit sphere, and then translated along a single axis (e.g., the x-axis) with a fixed margin. This preserves the intrinsic local geometry of each object while keeping them spatially distinct within a single 8,192-point input limit.

Table 6. Additional results on MO3D: scene-level and text-only baselines.

Model		Pos.	Comp.	Hol. (B)	Hol. (R)
Scene-level	Chat-Scene	10.8	3.0	49.7	31.6
	LL3DA (Click)	17.8	4.5	48.8	26.5
	LL3DA (Bbox)	18.2	2.9	48.8	25.9
Text-only	Vicuna-7B (GT captions)	52.0	28.9	70.5	50.3
Ours (Multi-3DLLM)		**56.3**	**33.8**	**81.7**	**57.2**

Scene-Level Models: We evaluated scene-level models, including Chat-Scene [11] and LL3DA [4]. Since these models are designed to extract objects from a full scene context, we provided them with the explicit centroid coordinates (click) or bounding boxes (BBox) of the target objects, mapping ordinal textual queries (*e.g.*, "the first object") to their corresponding spatial prompts. As shown in Table 6, despite this explicit localization, these models severely underperformed on MO3D and Shape Mating. This confirms our architectural finding: their object-level token pooling smooths out the fine-grained local geometry required for detailed comparison.

Text-Only Baseline (Language Bias): To isolate the contribution of 3D geometric reasoning from linguistic priors, we evaluated a text-only baseline using Vicuna-7B (Table 6). We provided the model with the ground-truth Cap3D text descriptions instead of visual inputs. The text-only model achieved scores on MO3D of 52.0 (Positional M), 28.9 (Comparative M), 70.5 (Holistic B), and 50.3 (Holistic R). While these scores demonstrate that the ground-truth captions offer a very strong semantic prior, our Multi-3DLLM consistently outperforms this text-only baseline across all metrics (e.g., 56.3 on Positional, 33.8 on Comparative). This confirms that our model's gains stem from genuine 3D geometry processing rather than merely exploiting language biases.

3.2. Architectural Ablation: Interaction Mechanics

Motivation. In the main paper, we demonstrated that the *Object-Level* interaction fails on geometric tasks. To further investigate whether this failure stems from the loss of salient features (due to mean pooling) or the loss of spatial resolution (due to object-wise broadcasting), we evaluate two additional architectural variants.

Variants.

- **Object-Level (Max Pooling):** Similar to the mean-pooling baseline, this variant aggregates object tokens into a single vector. However, it uses *max-pooling* to capture the most salient features (e.g., sharp corners or

handles) across the patch tokens. The updated residual is then broadcast uniformly to all patches of the object. This tests if preserving salient features is sufficient for geometric reasoning.

- **Micro-Token Interaction:** This variant operates at an intermediate granularity. Instead of collapsing an object into a single vector, we compress the object's patch tokens into $M = 32$ representative *"micro-tokens"* inspired by *0M-Pooling* from [20]. This is achieved by aggregating patch tokens into distinct clusters based on feature similarity, thereby reducing redundancy while preserving diverse local features. The interaction module processes these micro-tokens, and the update is redistributed to the original patches via a cross-attention mechanism, allowing for spatially varying updates.

Results and analysis The results in Table 7 offer a nuanced and critical insight. On the semantic comparison tasks (MO3D), both *Object-Level (Max)* and *Micro-Token* variants perform exceptionally well, slightly surpassing our model with PIT block (Hereafter PIT model). This suggests that for high-level semantic comparison, capturing salient features via max-pooling or representative tokens is sufficient. However, the results on Shape Mating reveal a fundamental limitation of these aggregation-based approaches. Both *Object-Level (Max)* and *Micro-Token* fail on this geometric task, scoring 23.9 and 24.5 on Selection (S), respectively. These scores are not only far below our *PIT model* but also worse than the *No-Interaction* baseline. This confirms that the failure of object-level models is not due to the pooling operation but stems from the architectural bottleneck of compressing local geometry into object-wise slots. Even with 32 micro-tokens, the spatial correspondence required for mating is lost.

3.3. Training Strategy

Motivation and Setup. Our training framework adopts a two-stage strategy. *Phase 1 (Feature Alignment)* follows the PointLLM methodology [21], training only the projector on the 660K brief-description instructions from Objaverse-Cap3D to align point cloud features with the LLM's embedding space. This is followed by *Phase 2 (Holistic Task-Mixture)*, which fine-tunes the Projector, PIT block, and LLM (while keeping the point encoder frozen) on our proposed benchmarks.

A natural question is whether Phase 1 is redundant: can the model learn to align modalities and reason about geometry simultaneously? To investigate this, we evaluate a *1-Stage* variant. In this setting, we initialize the projector randomly and train the full model (Encoder, Projector, and LLM) directly on the holistic data mixture. We compare this against our standard *2-Stage* approach.

Results and Analysis. The results in Table 9 reveal a critical trade-off between task-specific optimization and general reasoning capability. Interestingly, for our PIT model, the *1-Stage* approach yields surprisingly high scores on the Mini-Applications. It achieves a Selection score of 66.6 on Shape Mating and a Delta Captioning score of 56.0. However, this comes at a severe cost: performance on the main MO3D benchmark drops significantly. Specifically, the Positional score falls from 56.3 to 47.6, and the Comparative score decreases from 33.8 to 24.5. This suggests that without the initial alignment of Phase 1, the powerful PIT architecture tends to overfit to the specific templates and biases of the narrower Mini-App tasks, effectively becoming a task-specific specialist at the expense of general understanding. The *1-Stage* model learns to exploit the limited linguistic patterns of Shape Mating but fails to ground the diverse, open-ended concepts required for MO3D. In contrast, the *2-Stage* approach ensures that the model is first grounded in a broad 3D-text semantic space. This pre-alignment acts as a necessary foundation, preventing the model from collapsing into task-specific shortcuts and enabling the robust, generalized comparison capabilities shown in the MO3D results. Thus, Phase 1 is essential for training a true generalist 3D-LLM.

3.4. Robustness to Object Count (Scaling to 4–5 Objects)

While our standard dataset focuses on $n \in \{2, 3\}$ to maintain high token density for fine-grained geometric tasks, we investigated zero-shot extensions to scenes with 4 or 5 objects on the MO3D positional QA task. We evaluated two approaches:

Naive Scaling. When forcing $n = 4$ or 5 objects via Micro-Token compression to fit within the same fixed token budget, performance naturally drops on queries referencing the 4th or 5th objects (falling to 32% and 14%, respectively), indicating out-of-distribution difficulty and loss of fidelity.

Test-Time Coarse-to-Fine. Since many queries depend only on a small subset of objects, we apply a training-free, inference-time filtering strategy: (i) we extract referenced objects from the question via ordinal terms, (ii) add top-2 candidates using lightweight retrieval using CLIP [18], and (iii) remap the ordinals to this filtered subset before running Multi-3DLLM. This procedure successfully recovers reasoning capabilities, achieving 50% and 51% accuracy on 4-object and 5-object positional queries, respectively. This demonstrates a viable path for computation-efficient scaling despite LLM context limits.

4. Advanced Analysis on Shape Mating

4.1. Impact of Two-Turn Conversational Reasoning

In the main paper (Table 1), we employed a strict single-turn generation protocol for the Shape Mating task. The model was required to output both the pair selection and a detailed

Table 7. Extended ablation on interaction mechanics. We compare different pooling strategies (Mean vs. Max) and granularities (Object vs. Micro vs. Patch). *Micro-Token* uses 32 representative tokens per object. *w/ PIT* uses full patch-level interaction.

Interaction Mechanism	MO3D				Shape Mating		Change Captioning		
	Positional (%)	Comparative (%)	Holistic (%)		Selection (%)		Verify (%)		Delta Caption (%)
	M	M	B	R	S	R	B	R	M
No-Interaction (w/o PIT)	45.5	21.8	81.3	53.0	34.4	36.7	49.1	37.1	48.0
Object-Level (Mean)	52.9	32.3	81.0	49.7	25.0	23.7	**51.7**	34.7	49.6
Object-Level (Max)	56.6	36.5	81.3	45.1	23.9	22.5	50.9	37.2	50.0
Micro-Token ($M = 32$)	56.9	35.3	80.1	44.0	24.5	21.8	50.6	34.6	50.0
w/ PIT	56.3	33.8	**81.7**	**57.2**	**37.1**	**36.8**	51.2	**37.3**	**51.0**

geometric rationale in a single response (*e.g.*, "(1,3). Pair (1,3) can mate because..."). Under this constrained setting, Multi-3DLLM achieved a Selection accuracy of 37.1%.

However, forcing a combined output creates a well-known objective imbalance during training. The Cross-Entropy (CE) loss becomes dominated by the long, generative rationale tokens, which inadvertently penalizes the short, categorical selection tokens. To mitigate this, we evaluated the model using a *two-turn conversational (2-chat) approach*, inspired by the multi-turn capabilities of standard 3D-LLMs like PointLLM [21]. The task is decoupled as follows:

- **Turn 1 (Selection):** The user asks, "Which pairs can mate? select one that applies." The model responds strictly with the selection, *e.g.*, "(1,3)".
- **Turn 2 (Reasoning):** The user follows up with, "Explain why." The model then generates the geometric rationale.

By decoupling the objective, the model can dedicate its full attention to the geometric matching in the first turn without the loss being diluted by the generation of long explanations. We utilize this optimized two-turn protocol to explore the model's real-world robustness in the following transfer experiments.

4.2. Transfer to Real-World Scans

To demonstrate that Shape Mating is not merely a synthetic procedural task, we evaluated the zero-shot transfer capability of our model on real-world scanned datasets: ScanObjectNN and OmniObject3D.

A critical challenge in real-world transfer is the point density gap. Real-world scans often have severe point limitations (*e.g.*, ScanObjectNN is limited to $\sim$2048 points), creating a density bottleneck when fed into an encoder pre-trained on 8,192 points. To match point densities, we augmented the evaluation with OmniObject3D.

Using the highly effective *Two-Turn Conversational* prompt described above, we present the real-world transfer results in Table 8. When point densities are properly matched, our model achieves a viable zero-shot selection accuracy of 36.3% (well above the 25% chance level). Furthermore, with a brief fine-tuning on just 5K real-world samples, the performance surges to 62.0%. In stark contrast, the 2D-VLM baseline (LLaVA-7B) remains entirely at chance-level ($\sim$25%) across all settings. This confirms that the Shape Mating task fundamentally preserves its geometry-centric nature across domain shifts, and our architecture maintains its robustness in real-world scenarios.

Table 8. **Results on real-world scanned datasets.** Metric is **Selection Accuracy (S, %)** following the main paper. Chance rate is 25% for both.

Model	ScanObjectNN (Real)	OmniObject3D (Real)
LLaVA-7B (Zero-shot)	21.6	26.0
Ours (Zero-shot)	29.0	36.3
Ours (Fine-tune)	**48.0**	**62.0**

5. Evaluation Details

5.1. LLM-based Evaluation Prompts

To ensure a robust and semantic assessment, we utilize *GPT-4o-mini* as our primary evaluator. Unlike rigid n-gram metrics, this LLM-based judge can discern semantic equivalence and validate reasoning logic. We employ specific prompts for each metric type, as detailed below and illustrated in Figure 11.

Semantic Accuracy (M) for MO3D. For open-ended questions in MO3D, exact string matching is insufficient. Our evaluation prompt instructs the judge to rate a response as Correct (1) or Incorrect (0). Crucially, this metric incorporates Visual Grounding. The evaluator is provided with both the ground-truth text and the multi-view images of the point clouds. It is instructed to accept the model's answer if it: (1) matches the ground truth semantically, or (2) provides a valid alternative description that is clearly supported by the visual evidence in the images, even if it differs from the text.

Semantic Accuracy (M) for Delta Captioning. For the Change Captioning (Delta) task, a binary score is too coarse. We employ a 10-point scale prompt. The evalua-

Table 9. Ablation on training stages across architectures. We compare the *1-Stage* and *2-Stage* (Alignment → Holistic Tuning) strategies for both the *No-Interaction* baseline and full model *w/ PIT*. The results investigate whether the initial feature alignment (Phase 1) is universally beneficial or specifically critical for our patch-interaction mechanism.

Model & Training Strategy	MO3D				Shape Mating		Change Captioning		
	Positional (%)	Comparative (%)	Holistic (%)		Selection (%)		Verify (%)		Delta Caption (%)
	M	M	B	R	S	R	B	R	M
No-Interaction (1-Stage)	47.6	21.0	78.9	41.9	23.8	22.0	49.1	39.4	42.0
No-Interaction (2-Stage)	45.5	21.8	81.3	53.0	34.4	36.7	49.1	37.1	48.0
w/ PIT (1-Stage)	47.6	24.5	78.9	44.8	**66.6**	**64.1**	50.3	38.4	**56.0**
w/ PIT (2-Stage)	**56.3**	**33.8**	**81.7**	**57.2**	37.1	36.8	**51.2**	**37.3**	51.0

tor decomposes the ground-truth edit instruction into atomic components (*e.g.*, "thicker legs", "higher back") and grades the generated description based on the recall of these components. Contradictions (*e.g.*, describing "thinner legs" when the truth is "thicker") result in an immediate score of 0.

Reasoning Accuracy (R). For tasks requiring justification (Shape Mating, Change Captioning, and MO3D Holistic), we evaluate the quality of the "Why" output. The prompt provides the judge with the context (objects/instruction), the model's selected answer, and its reasoning. The judge assigns a score of 1 only if the reasoning is logically sound, factually consistent with the ground truth answer.

Validation of the LLM Judge. To ensure the LLM metric is a reliable proxy for semantic evaluation, we conducted a human audit (3 people) of 300 randomly selected LLM-judged responses. The LLM's decisions achieved 91.0% unanimous human support (and 98.0% with at least one human vote). This confirms that our evaluation protocol accurately reflects human judgment in assessing 3D geometric descriptions and reasoning.

5.2. Standard NLP Metrics

We report standard NLP metrics *BLEU-4* [16], *ROUGE-L* [9, 13], *METEOR* [3], *SimCSE* [10] for all generative tasks. Table 11 presents the results for the Mini-Applications (Change Captioning and Shape Mating), and Table 10 presents the results for the MO3D dataset. For 2D-VLMs, scores are explicitly reported for both 1-view and 2-view settings.

Analysis of Metric Discrepancies. We observe instances where standard NLP metrics diverge from our semantic evaluators. For example, in Shape Mating, 2D-VLMs achieve high SimCSE scores (*e.g.*, LLaVA: 67.65) despite near-zero Selection accuracy. This indicates "hallucinated fluency": generating plausible-sounding but geometrically incorrect text. Similarly, in MO3D Positional, the *w/o PIT* baseline slightly edges out full Multi-3DLLM on BLEU-4, yet fails significantly on Semantic Accuracy (M). This suggests the baseline relies on memorizing safe linguistic patterns, whereas our model generates more diverse, geometrically grounded descriptions that differ from the ground truth text but are verified as correct by the LLM judge. These discrepancies underscore the necessity of our proposed LLM-based metrics for accurate benchmarking.

6. Additional Qualitative Results

We provide extensive qualitative examples to visually demonstrate the capabilities and limitations of our model compared to state-of-the-art baselines.

Comparison on MO3D (Main Task). Figures 12 to 16 (Examples 1–5) present results comparisons on the MO3D benchmark.

Performance on Mini-Applications. Figures 17 to 20 (Examples 6–9) showcase results on the application-driven benchmarks.

Table 10. **Standard NLP Metrics for MO3D.** Detailed scores for Positional, Comparative, and Holistic QA tasks. SimCSE scores evaluate semantic similarity.

Model	Positional QA				Comparative QA				Holistic QA			
	B-4↑	R-L↑	MET↑	Sim↑	B-4↑	R-L↑	MET↑	Sim↑	B-4↑	R-L↑	MET↑	Sim↑
LLaVA (1-view)	23.97	49.84	47.43	62.31	3.97	26.48	28.66	58.46	2.19	20.03	23.05	57.68
LLaVA (2-view)	23.79	49.55	47.48	61.62	3.59	26.18	27.52	58.16	2.29	20.32	23.51	58.19
Molmo (1-view)	3.88	24.27	31.26	55.04	3.35	21.26	27.87	63.21	1.01	15.15	19.92	54.22
Molmo (2-view)	3.40	24.20	30.70	55.06	3.40	21.23	27.43	62.79	0.99	14.92	19.17	53.47
MiniGPT-3D	15.99	44.01	50.36	63.04	13.73	42.01	40.52	64.59	4.20	24.38	24.28	58.46
PointLLM	29.95	55.60	52.22	64.65	16.06	43.75	41.77	64.13	3.60	21.32	21.97	61.26
ShapeLLM	31.08	56.67	53.86	63.77	24.23	49.68	48.56	71.43	4.25	20.85	20.76	62.82
Multi-3DLLM (w/o PIT)	**46.83**	**69.52**	66.44	**79.56**	40.74	62.06	62.96	80.14	**26.87**	**52.34**	**49.84**	**76.22**
Multi-3DLLM (Ours)	45.54	68.95	**66.59**	78.32	**40.99**	**62.83**	**63.38**	79.72	22.79	49.25	47.40	74.52

Table 11. **Standard NLP Metrics for Mini-Applications.** Detailed scores for Shape Mating, Change Captioning (Verify), and Change Captioning (Delta).

Model	Shape Mating				Change Captioning (Verify)				Change Captioning (Delta)			
	B-4↑	R-L↑	MET↑	Sim↑	B-4↑	R-L↑	MET↑	Sim↑	B-4↑	R-L↑	MET↑	Sim↑
LLaVA (1-view)	1.17	21.99	15.31	67.48	4.53	30.05	23.57	56.27	3.58	26.69	19.85	55.86
LLaVA (2-view)	1.17	22.23	15.42	**67.81**	4.52	28.51	22.76	55.39	3.33	26.26	18.77	56.28
Molmo (1-view)	2.59	24.66	20.60	57.78	1.98	21.11	24.29	55.84	1.52	18.26	21.49	52.71
Molmo (2-view)	2.85	23.26	20.60	60.45	2.02	21.50	24.77	55.76	1.49	18.24	21.73	52.78
MiniGPT-3D	1.91	14.93	15.78	45.86	2.60	22.12	15.87	45.47	0.98	11.95	7.76	33.75
PointLLM	1.13	13.04	12.61	42.74	1.76	14.22	13.62	38.29	0.72	11.90	8.88	39.90
ShapeLLM	1.73	16.58	13.47	43.70	1.33	9.60	10.44	23.68	1.23	15.09	13.92	48.21
Multi-3DLLM (w/o PIT)	16.28	31.43	28.19	53.28	10.40	34.67	29.20	55.66	5.00	25.98	**24.14**	**68.42**
Multi-3DLLM (Ours)	**16.65**	**31.40**	**28.37**	53.23	**12.47**	**37.67**	**30.76**	**57.62**	**5.02**	**26.59**	23.98	67.60

System Prompt for MO3D QA Generation

You are an AI assistant creating a high-quality dataset for a 3D vision-language model. Your task is to generate question{answer pairs for three task types (positional, comparative, holistic) based on the object descriptions below. (and supplementary multi-view images when available). Use visual evidence as the primary source for visual attributes. Use descriptions to supplement non-visual semantics. When images contradict the descriptions on visual attributes, trust the images. Do not claim uniqueness from omission in descriptions; verify across images.

Input Context: *[Object Descriptions & Multi-view Images]*

CRITICAL INSTRUCTIONS:

1. Produce exactly two distinct question{answer pairs for each task type.
2. Answers must be direct, factual statements grounded in the images.
3. Do NOT use object category nouns; instead refer to \the first object", etc.
4. Do NOT use spatial relations (left/right/front/behind).
5. Comparative QA must emphasize structural or functional differences.
6. Holistic QA must include one \Yes" and one \No" answer.
7. MANDATORY Category-Specific Questions: All questions must explicitly reference the target category (e.g., {target_category}). The question text must use the category's REQUIRED KEYWORDS to avoid ambiguity.

   REQUIRED KEYWORDS (Excerpt):
   -- Geometry / Structure: MUST use: \shape", \form", \geometric", \structural design" FORBIDDEN: \feature", \property", \appearance"
   -- Material: MUST use: \material", \made of", \constructed from" FORBIDDEN: \feature", \property", \what is X"
   -- *[Additional mandatory keyword rules for: Color, Function, Taxonomy, Style/Aesthetics]*

8. Within each task, the two variants must rely on different properties.
9. Do NOT assume a property is unique unless the images clearly show uniqueness.
10. For positional questions referencing a \unique" feature, ensure exactly one object has that feature.

Task Definitions:
-- positional_qa: Question about one object's attribute.
-- comparative_qa: Question comparing structural or functional aspects.
-- holistic_qa: Question about a property shared (or not shared) by all objects.

Figure 5. **System Prompt for MO3D QA Generation.** To ensure high-quality, non-ambiguous questions, we enforce strict keyword constraints (Instruction 7) for each target category. For readability, we list representative keywords for the "Geometry" and "Material" categories; identical constraint logic is applied to the other categories (Color, Function, Taxonomy, Style).

System Prompt for Shape Mating Rationale Paraphrasing

```
You are rewriting rationales for a 3D part mating QA dataset. Three interface parts
are labeled (1), (2), and (3). The correct mating pair list must remain [Answer
List]. The canonical rationales are: [Canonical Rationales]
Task: Paraphrase each rationale in fresh wording while keeping the facts.
Guidelines:
1. Preserve the logical meaning of every rationale.
2. Mention complementary vs. conflicting geometry explicitly.
3. Keep each entry to one or two sentences. Vary phrasing in a [Style Hint] style.
4. Do not introduce new geometry details or contradict the canonical text.
5. Return JSON only, with keys 'answer' and 'why'.
   • 'answer' must be the same list of mating pairs.
   • 'why' must map each pair key to your rewritten rationale.
6. Use the exact keys '(1,2)', '(1,3)', '(2,3)'.
7. No code fences, no additional commentary.
```

Figure 6. **Full Prompt for Shape Mating Rationale Generation.** We utilize GPT-4o-mini to convert structured error tags (e.g., `cut_mismatch`) into natural language explanations.

Templates for Change Captioning Tasks

Task 1: Verification (Binary Classification)

Input: Anchor Point Cloud (P_A), Candidate Point Cloud (P_C), Instruction (I)
Randomization: The order of input point clouds (P_A, P_C) is randomized.

[Template A: Anchor is First]
Input: <point> (P_A) <point> (P_C)
Q: Does the second object satisfy all of the following requirements compared to the first object?
Requirements: - [Instruction]

[Template B: Anchor is Second]
Input: <point> (P_C) <point> (P_A)
Q: Does the first object satisfy all of the following requirements compared to the second object?
Requirements: - [Instruction]

Task 2: Delta Captioning (Generative)

Input: Anchor Point Cloud (P_A), Positive Point Cloud (P_P)
Goal: Generate a description of the geometric edit.

[Templates (Randomly Selected)]
- How would you transform the first object so that it matches the second object?
- Describe the edits needed to convert the first object into the second.
- What modifications should be applied to the first object to obtain the second?
- List the geometric adjustments required to turn the first object into the second.

Figure 7. **Prompt Templates for Change Captioning.** We utilize a set of diverse templates for the Delta Captioning task. For the Verification task, we explicitly randomize the input order of the anchor and candidate objects and adjust the question wording (”first” vs ”second” object) accordingly to prevent positional bias.

Table 12. **Instruction Templates for Shape Mating.** We utilize 15 variations of the prompt to train the model, ensuring robustness to phrasing while maintaining a consistent task definition (Selection + Explanation).

Template Variations
• Which pairs can mate? select one that applies. Options: (1,2), (1,3), (2,3), None Explain why each chosen pair can mate and why others cannot.
• Identify which pairs can mate. Choose the applicable option. Options: (1,2), (1,3), (2,3), None Provide explanations for why each selected pair can mate and why the others cannot.
• Determine which pairs are able to mate. select one that applies. Options: (1,2), (1,3), (2,3), None Explain the reasoning for each pair that can mate and why the remaining pairs cannot.
• Find the pairs that can mate together. Select the applicable option. Options: (1,2), (1,3), (2,3), None Describe why each chosen pair can mate and explain why the other pairs cannot.
• Which pairs can successfully mate? select one that applies. Options: (1,2), (1,3), (2,3), None Explain why each selected pair can mate and provide reasons why the others cannot.
• Determine all pairs that are capable of mating. Select the applicable options. Options: (1,2), (1,3), (2,3), None Provide explanations for why each selected pair can mate and why others cannot.
• Which pairs can mate with each other? select one that applies. Options: (1,2), (1,3), (2,3), None Explain why each chosen pair can mate and describe why the other pairs cannot.
• What pairs are able to mate? select one that applies. Options: (1,2), (1,3), (2,3), None Explain why each selected pair can mate and justify why the other pairs cannot.
• Find all pairs that can mate. Choose applicable option. Options: (1,2), (1,3), (2,3), None Describe why each chosen pair can mate and explain why the remaining pairs cannot.
• Which pairs can be mated together? select one that applies. Options: (1,2), (1,3), (2,3), None Explain why each selected pair can mate and provide reasoning for why others cannot.
• Identify the pairs that can mate. Choose the applicable option. Options: (1,2), (1,3), (2,3), None Explain why each chosen pair can mate and describe why the other pairs cannot.
• Determine which pairs are compatible for mating. select one that applies. Options: (1,2), (1,3), (2,3), None Provide explanations for why each selected pair can mate and why the remaining pairs cannot.
• What pairs can successfully mate? select one that applies. Options: (1,2), (1,3), (2,3), None Explain why each chosen pair can mate and justify why others cannot.
• Which pairs are capable of mating? select one that applies. Options: (1,2), (1,3), (2,3), None Explain why each selected pair can mate and provide reasons why the other pairs cannot.

Table 13. **Instruction Templates for Change Captioning.** We use distinct template sets for the Verification and Delta Captioning tasks. For Verification, we explicitly randomize the object order (Anchor first vs. second) to prevent positional bias.

Task Type	Template Variations
Verification	• *(Condition: Anchor is First object):* Does the second object satisfy all of the following requirements compared to the first object? Requirements: - [Instruction] • *(Condition: Anchor is Second object):* Does the first object satisfy all of the following requirements compared to the second object? Requirements: - [Instruction]
Delta Captioning	• How would you transform the first object so that it matches the second object? • Describe the edits needed to convert the first object into the second. • What modifications should be applied to the first object to obtain the second? • List the geometric adjustments required to turn the first object into the second.

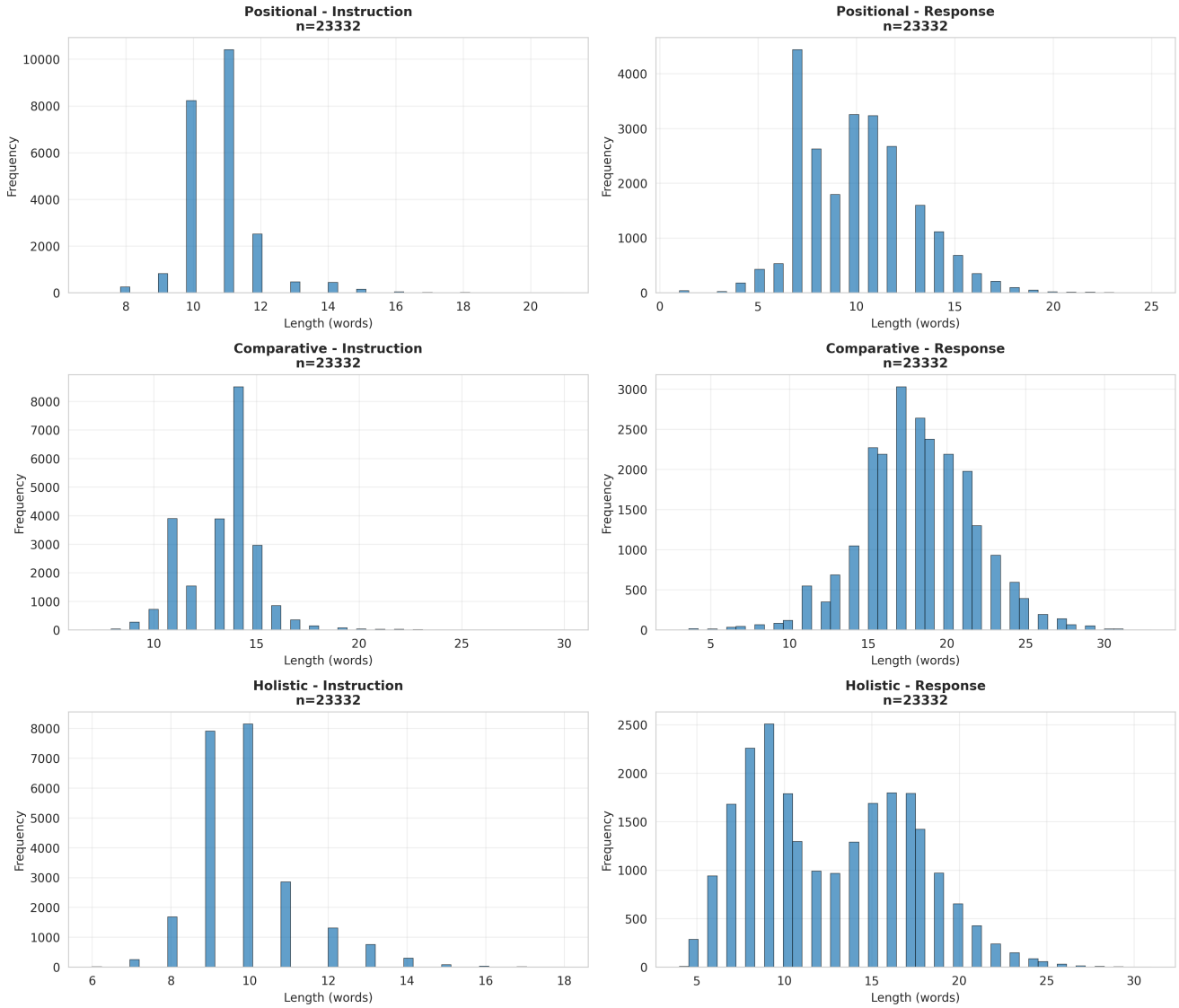


Figure 8. **Distribution of instruction and response lengths in MO3D dataset.**

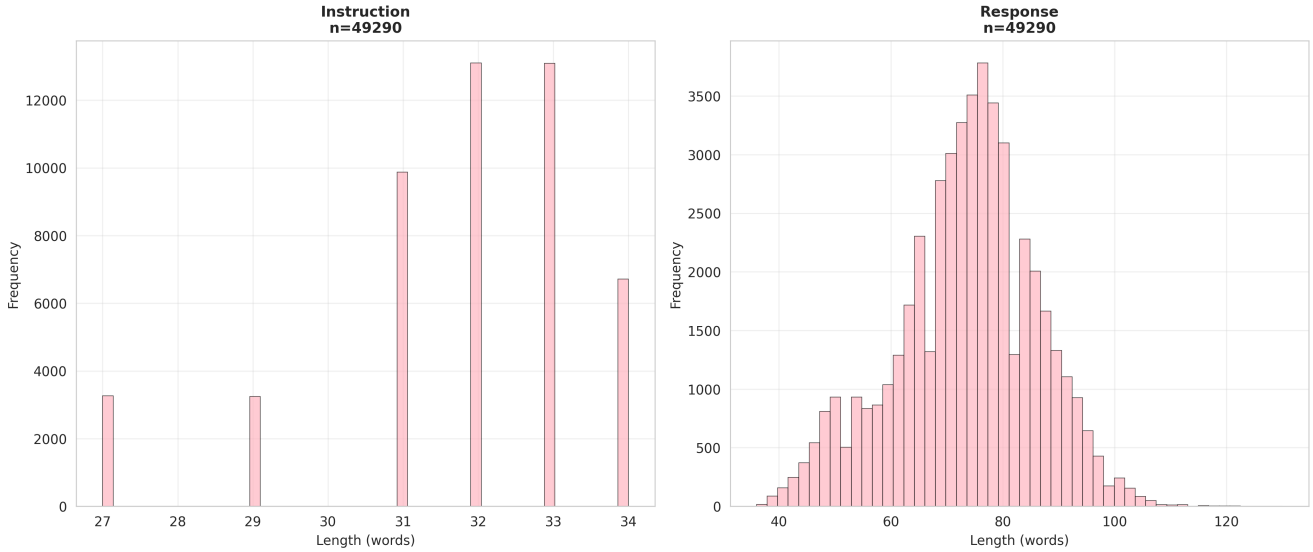


Figure 9. **Distribution of instruction and response lengths in Shape Mating.**

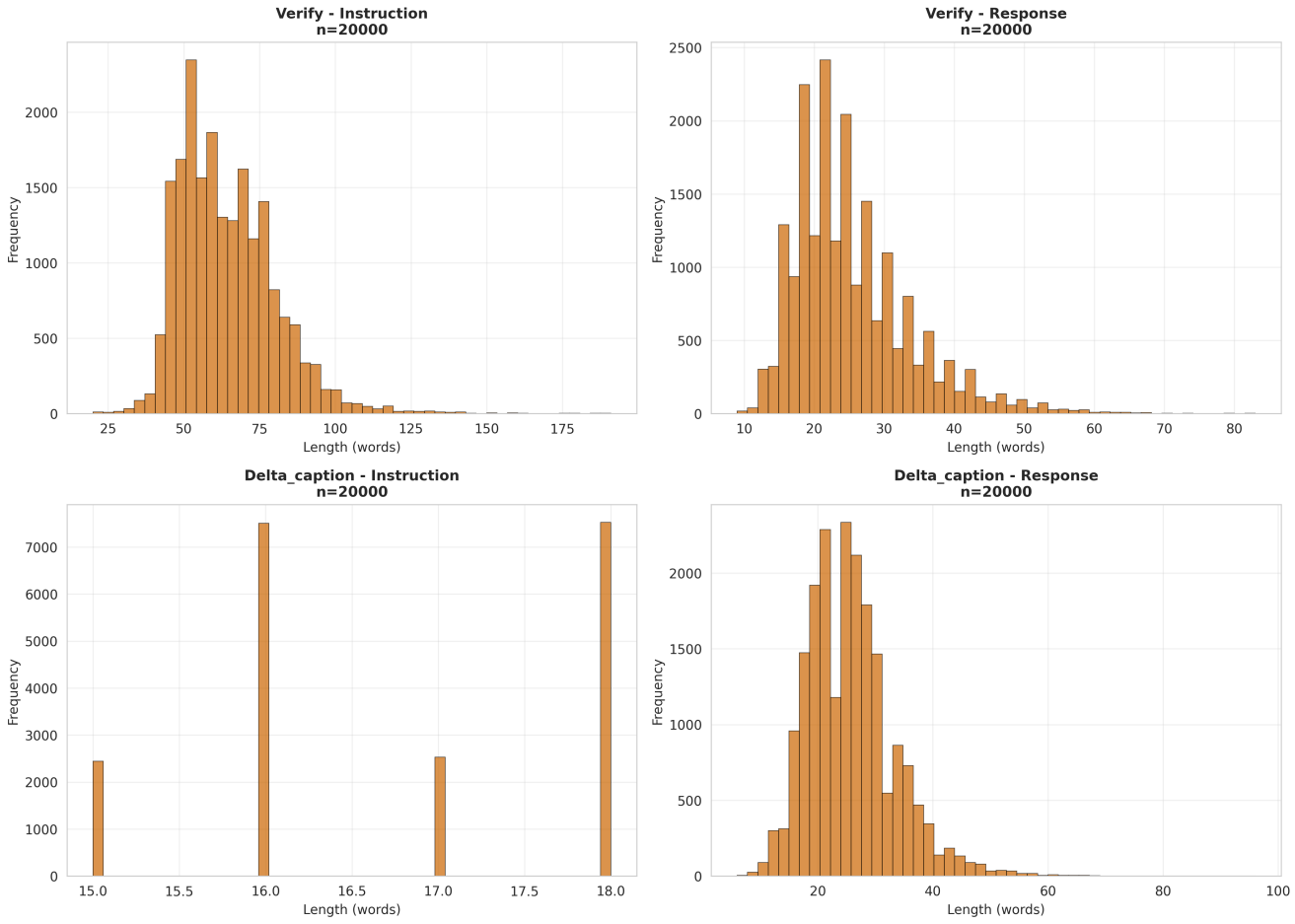


Figure 10. **Distribution of instruction and response lengths in Change Captioning.**

LLM Evaluation Prompts (GPT-4o-mini)

[Metric M: Semantic Accuracy for MO3D]

```
System: You are an impartial grader for a 3D-QA benchmark with multi-view image
evidence.
CRITICAL: The images are the PRIMARY source of truth. Ground-truth text is a
reference.
Criteria:
1. Accept the answer if it matches the ground-truth text semantically.
2. Accept the answer if it reasonably describes what is visible in the images, even if
   it differs from the ground-truth text (e.g., specific material/shape details).
3. Only reject if the answer clearly contradicts what is visible in ALL provided
   images.
Output JSON: {"score": 1 or 0, "reason": "..."}
```

[Metric M: Delta Captioning Score (10-point scale)]

```
System: Evaluate the model's description of geometric changes using a 10-point scale.
Instructions:
1. Break down the ground truth into individual geometric modification items.
2. Check how many items are captured by the model.
3. If the model contradicts any item, return M=0.
4. Score based on coverage: 10 (All items correct), 7-9 (Most correct), 4-6 (Half
   correct), 1-3 (Few correct), 0 (Contradiction/None).
```

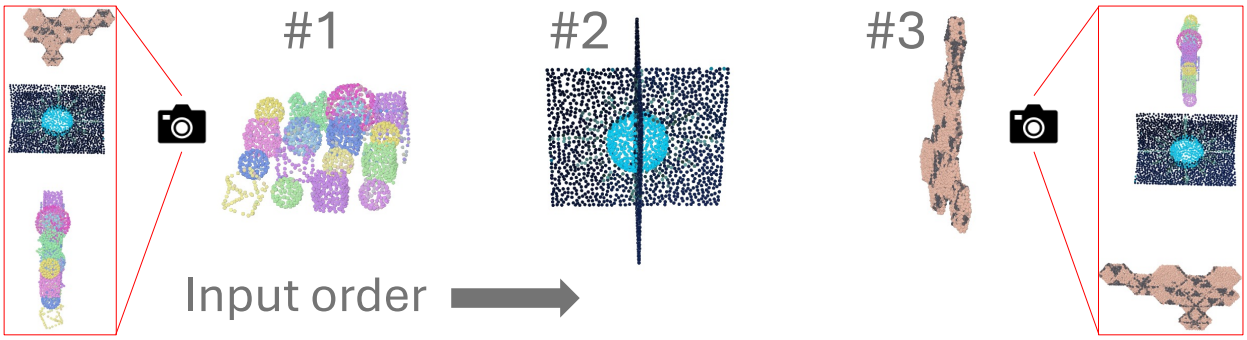
Output JSON: {"M": 0-10, "reason": "..."}

[Metric R: Reasoning Accuracy]

```
System: Evaluate whether the model's reasoning is factually consistent with the
requirements and ground truth.
Criteria:
• Is the reasoning logically sound?
• Does it correctly justify the selected answer/conclusion?
Output JSON: {"R": 1 or 0, "reason": "..."}
```

Figure 11. **System Prompts for Evaluation.** We use specific prompts for different metric types. For MO3D (Top), the evaluator is explicitly instructed to prioritize visual evidence from multi-view renderings over text labels to handle valid open-ended descriptions.

Example 1: Positional Understanding

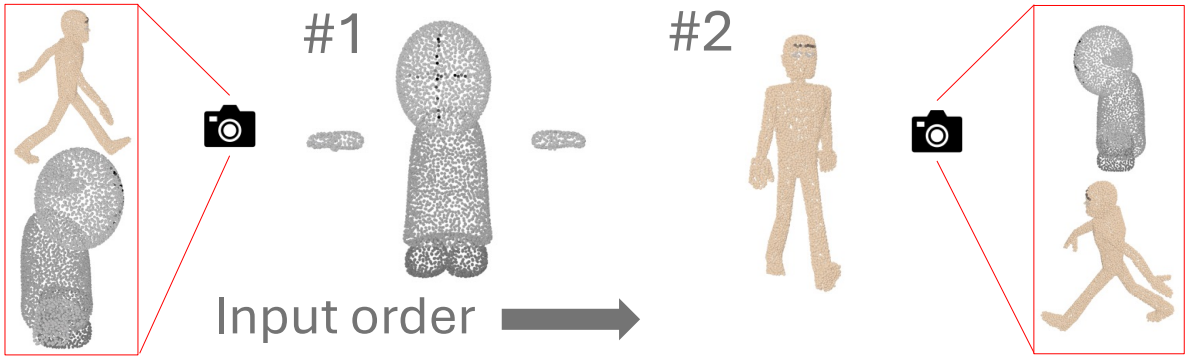


Question: What is the geometric shape of the third object?
Ground Truth: The third object has a hexagonal geometric shape.

Model	Response
LLaVA (1-view) [14]	The third object is a square.
Molmo (1-view) [8]	The geometric shape of the third object is a cross. This cross shape is formed by four rectangular prisms extending outward from a central point, creating a symmetrical cross-like structure.
MiniGPT-3D (Concat) [19]	the third object is a cube.
PointLLM (Concat) [21]	The third object is in the shape of a cube.
ShapeLLM (Concat) [17]	The third object is a cube.
Multi-3DLLM (Ours)	**The third object has a hexagonal geometric shape.**

Figure 12. **Qualitative Comparison on MO3D (Positional Task).**

Example 2: Positional Understanding



Question: What is the structural design of the second object?
Ground Truth: The second object has a humanoid form with limbs positioned in a walking pose.

Model	Response
LLaVA (1-view) [14]	The second object is a cartoon-like figure, which is a representation of a person walking.
Molmo (1-view) [8]	The second object in the image is a human figure with a unique structural design. It has a simple, stylized form with a round head and a single, continuous line that forms the body, arms, and legs. The figure is depicted in a walking pose, with the left leg extended forward and the right leg behind. The arms are positioned in a way that suggests movement, with the left arm bent at the elbow and the right arm extended backward. The figure's skin tone is a light brown color, and there's a small black area above the left eye, which could represent an eye or some other facial feature. This design creates a minimalist yet dynamic representation of a human in motion.
MiniGPT-3D (Concat) [19]	the second object is a humanoid figure with a distinct, elongated torso and a head that is significantly larger than the rest of its body.
PointLLM (Concat) [21]	The second object is designed to resemble an arrow.
ShapeLLM (Concat) [17]	The second object has a cylindrical body with a flat base and two protruding arms or legs.
Multi-3DLLM (Ours)	**The second object has a humanoid form with outstretched arms.**

Figure 13. **Qualitative Comparison on MO3D (Positional Task).**

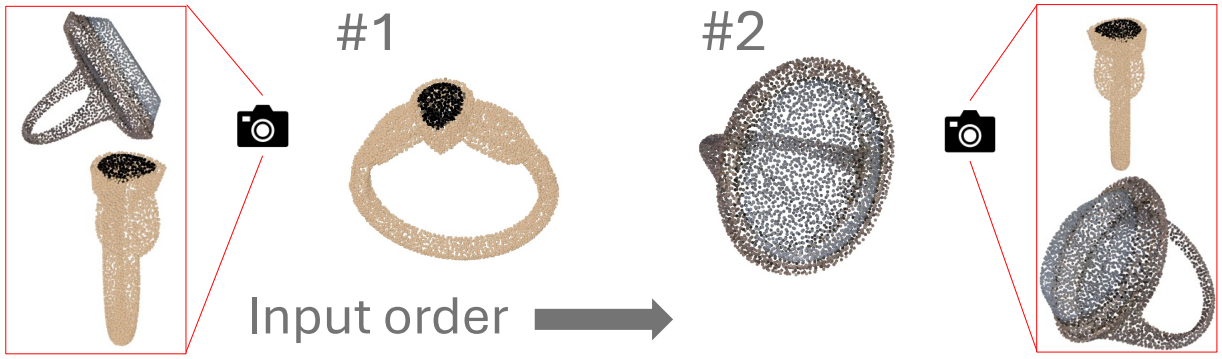
Example 3: Comparison Understanding

Question: How do the colors of the first and third objects compare?
Ground Truth: The first object is primarily blue and green, while the third object features a camouflage pattern with green and brown shades.

Model	**Response**
LLaVA (1-view) [14]	The first object, which is a green hat, has a similar color to the third object, which is a green shirt. The hat and shirt both have a green hue, making them visually similar. The second object, a blue hat, has a distinct color compared to the green hat and shirt, as it is a different shade of blue.
Molmo (1-view) [8]	The first and third objects in the image share the same color scheme. Both are wearing green pants. The first object has a green and yellow shirt, while the third object has a green shirt with yellow and red accents. So while the shirts differ in their patterns and additional colors, the base color for both the pants and shirts in the first and third objects is green.
MiniGPT-3D (Concat) [19]	the first object, a green hat, contrasts with the blue shirt of the second object.
PointLLM (Concat) [21]	The first and third objects have similar colors - both are green and blue.
ShapeLLM (Concat) [17]	The first object is green, while the third object is blue.
Multi-3DLLM (Ours)	**The first object has a green hat and blue glasses, while the third object is primarily green with yellow accents.**

Figure 14. **Qualitative Comparison on MO3D (Comparison Task).**

Example 4: Holistic Understanding



Question: Do both objects have a central gem or lens?
Ground Truth: Yes. Both have a central element, one being a black diamond and the other a clear glass lens.

Model	Response
LLaVA (1-view) [14]	Yes, both objects have a central gem or lens.
Molmo (1-view) [8]	No. Only the left object has a central gem or lens. The right object appears to be a simple ring without any central feature.
MiniGPT-3D (Concat) [19]	no, only the ring has a central gem.
PointLLM (Concat) [21]	Yes, both objects have a central gem or lens
ShapeLLM (Concat) [17]	Yes, both objects have a central gem or lens.
Multi-3DLLM (Ours)	**Yes. Both objects feature a central diamond or lens.**

Figure 15. **Qualitative Comparison on MO3D (Holistic Task).**

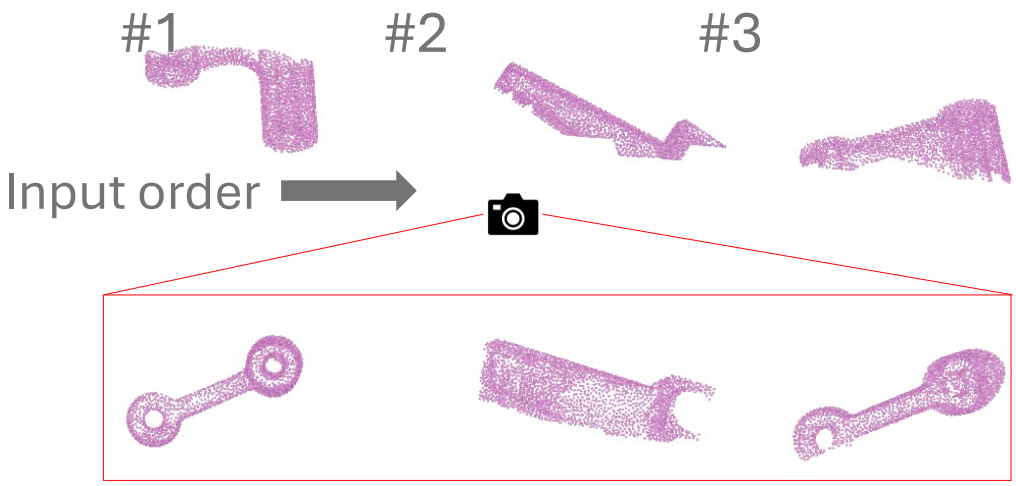
Example 5: Holistic Understanding

Question: Do all objects have a cylindrical form?
Ground Truth: No. Only the second object has a cylindrical form, while the first object is a bottle and the third object is a canister.

Model	Response
LLaVA (1-view) [14]	No, not all objects have a cylindrical form. In the image, there is a bottle, a can, and a vase, which are all cylindrical in shape. However, there is also a box, which is not cylindrical.
Molmo (1-view) [8]	No, not all objects in the image have a cylindrical form. While there is a prominent cylindrical structure in the center, which appears to be a stone or concrete pillar with a square top, there are other objects visible that have different shapes. In the background, you can see a rectangular object, and there's also a triangular shape visible. These various shapes create an interesting contrast with the main cylindrical structure, adding depth and diversity to the overall composition of the image.
MiniGPT-3D (Concat) [19]	no, not all objects have a cylindrical form. the objects in the 3d model are a bottle, and a can.
PointLLM (Concat) [21]	No, while two objects do have a cylindrical form, the third one is in the shape of a cuboid.
ShapeLLM (Concat) [17]	No, not all objects have a cylindrical form.
Multi-3DLLM (Ours)	**Yes. All objects exhibit a cylindrical shape.**

Figure 16. **Qualitative Comparison on MO3D (Holistic Task).** (Note: This is a failure case where our model incorrectly generalizes the cylindrical shape to all objects.)

Example 6: Shape Mating (Selection)

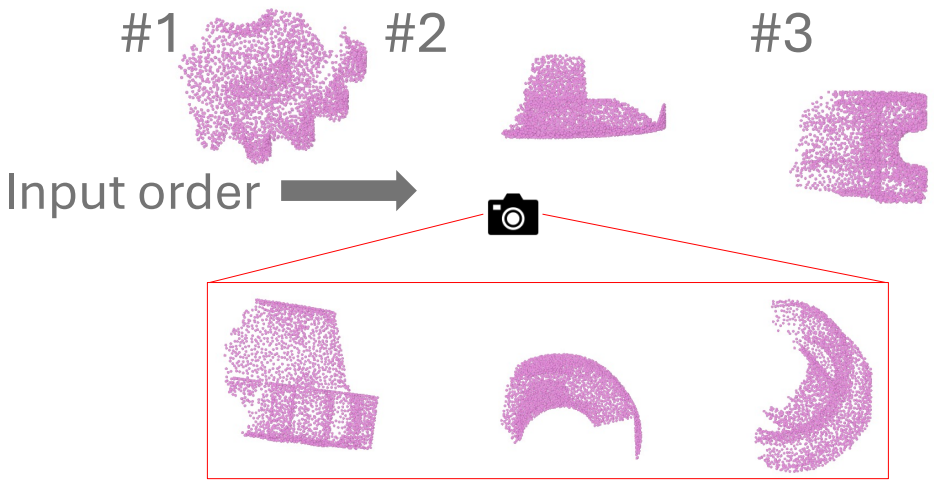


Question: Which pairs can mate? Select one that applies. Options: (1,2), (1,3), (2,3), None. Explain why each chosen pair can mate and why others cannot.
Ground Truth: Pair (1,3) can mate because these pieces were designed as complementary halves, allowing their pulsed ridge features to interlock tightly.

Model	Response
LLaVA (1-view) [14]	The pair that can mate is (1,2) because the two shapes are adjacent to each other, and they share a common edge. In contrast, the other pairs (1,3) and (2,3) cannot mate because they do not share a common edge, and the shapes are not adjacent to each other.
Molmo (1-view) [8]	The pair that can mate is (1,2). Pair (1,2) can mate because the jagged, irregular shape on the left resembles a shark, while the shape on the right looks like a fish. Sharks and fish are both aquatic animals and can potentially interbreed in certain marine ecosystems. Pairs (1,3) and (2,3) cannot mate because the shapes associated with these pairs do not resemble compatible species. The jagged shape for pair (1) and the irregular shape for pair (2) are not recognizable as any known animal types that could potentially breed. None of the other pairs show shapes that could represent compatible species for mating. The abstract nature of the shapes makes it impossible to identify any potential breeding relationships for pairs (2,3) or (1,3). Therefore, based on the visual representations provided, the only possible mating pair is (1,2), with the jagged shape resembling a shark and the irregular shape resembling a fish.
MiniGPT-3D (Concat) [19]	the pairs that can mate are (1,2) and (2,3) because they share a common feature, a hole in the middle. the other pair (1,3) cannot mate because they do not share a common feature.
PointLLM (Concat) [21]	The pair (1,2) can mate as they are of the same species, while the pair (1,3) and (2,3) cannot as they are of different species.
ShapeLLM (Concat) [17]	The pair (1,2) can mate as they are adjacent to each other, while the pair (1,3) cannot mate as they are not adjacent to each other.
Multi-3DLLM (Ours)	**(1,3). Pair (1,3) can mate because their mating faces share matching pulsed interface contours with mirrored normals, allowing them to seat without gaps.**

Figure 17. **Qualitative Comparison on Shape Mating (Selection Task).**

Example 7: Shape Mating (Selection)

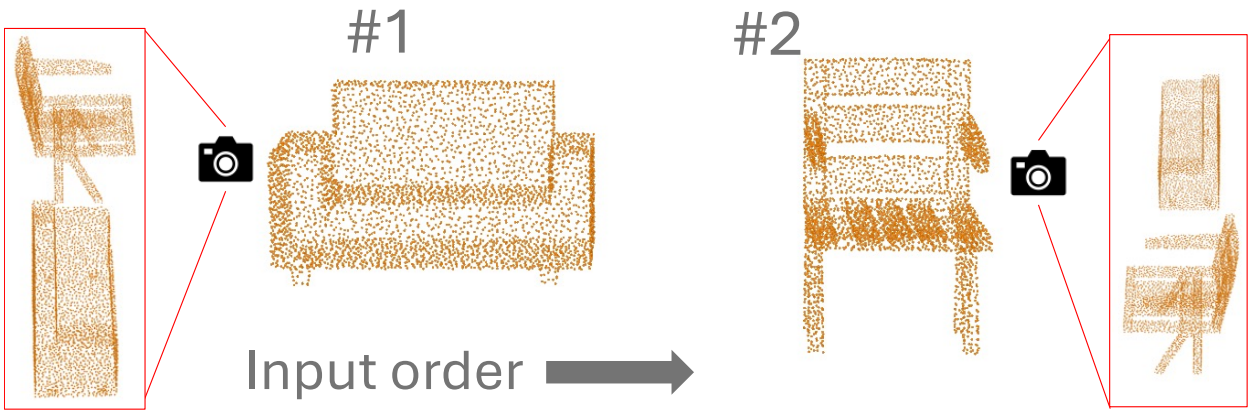


Question: Find the pairs that can mate together. Select the applicable option. Options: (1,2), (1,3), (2,3), None. Describe why each chosen pair can mate and explain why the other pairs cannot.
Ground Truth: The correct mating pair (2,3). Pair (2,3) mates because their mating faces share matching stepped square interface contours with mirrored normals, allowing the surfaces to nest snugly.

Model	Response
LLaVA (1-view) [14]	The pair that can mate is (1,2) because the two shapes are connected and share a common edge. In contrast, the other pairs (1,3) and (2,3) are not connected and do not share any common edges, making them unable to mate.
Molmo (1-view) [8]	the pair that can mate is (1,2). Option 1: The first object, which resembles a keyhole, can mate with option 2. This pairing is possible because both shapes have similar circular openings, suggesting they could fit together or complement each other in some way. Option 2: The second object, which looks like a key, can mate with option 1. The key's shape is complementary to the keyhole, with its pointed tip and shaft potentially fitting into the circular opening of the keyhole. Options 1 and 2 can mate because their shapes are designed to fit together, with the key's tip potentially aligning with the keyhole's opening. Option 3: The third object, which appears to be a different shape entirely, cannot mate with either 1 or 2. Its distinct form means it doesn't have the necessary compatibility to fit into or work with the keyhole or key shapes. None: The fourth option, being non-existent, cannot mate with any other pair as it doesn't represent a real object that could potentially fit or work in conjunction with the keyhole or key.
MiniGPT-3D (Concat) [19]	the pairs that can mate are (1,2) and (2,3) because they share a common feature, a hole in the middle. the other pair (1,3) cannot mate because they do not share a common feature.
PointLLM (Concat) [21]	The pair (1,2) can mate as they are of the same size and shape, whereas the pair (1,3) cannot as they have different shapes. The pair (2,3) also cannot as they have different sizes.
ShapeLLM (Concat) [17]	The pair (1,2) can mate as they are adjacent to each other, while the pair (1,3) cannot mate as they are not adjacent to each other
Multi-3DLLM (Ours)	**(2,3) Pair (2,3) can mate because their mating faces share matching stepped square interface contours, allowing them to seat without gaps.**

Figure 18. **Qualitative Comparison on Shape Mating (Selection Task).**

Example 8: Change Captioning (Verify)

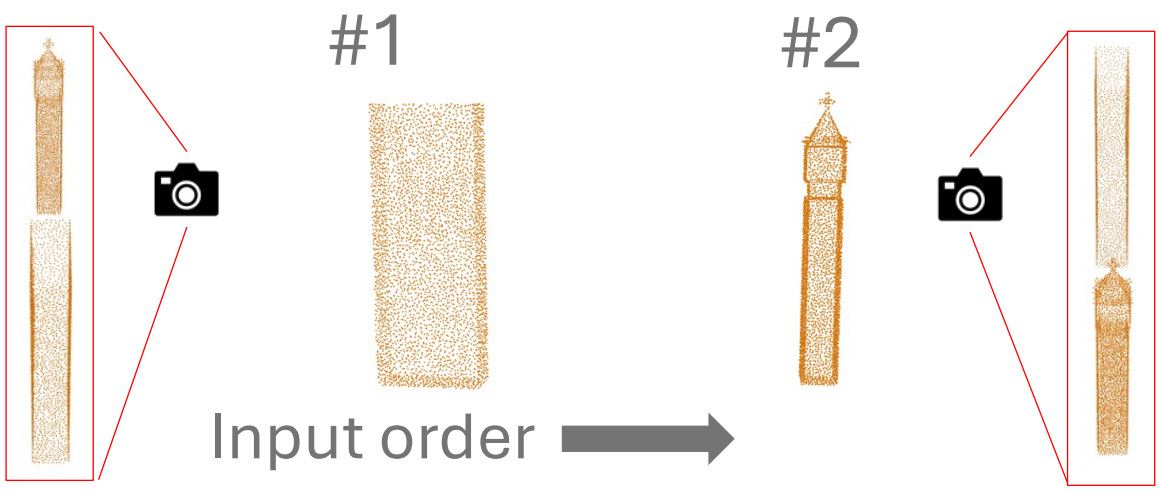


Question: Does the first object satisfy all of the following requirements compared to the second object? Requirements: The back rest is taller - The legs are longer - The seat is square - The back rest is not as wide - It does not have arm rests
Ground Truth: No. While the legs are longer, the back rest is not narrower and does not meet all the specified requirements.

Model	**Response**
LLaVA (1-view) [14]	Yes, the first object satisfies all of the requirements compared to the second object. The first object has a taller back rest, longer legs, and a square seat, while the second object has a shorter back rest, shorter legs, and a round seat. Additionally, the first object does not have arm rests, which is another difference between the two objects.
Molmo (1-view) [8]	The first object does not satisfy all of the requirements compared to the second object. Here's a breakdown: - The back rest is taller: The first object has a taller back rest than the second object. - The legs are longer: The first object's legs are longer than those of the second object. - The seat is square: Both objects have square seats, so this requirement is met. - The back rest is not as wide: The first object's back rest is wider than the second object's.- It does not have arm rests: The first object does not have arm rests, while the second object does. In summary, the first object meets 3 out of 5 requirements compared to the second object. It has a taller back rest, longer legs, and a wider seat. However, it lacks arm rests and has a narrower back rest in comparison
MiniGPT-3D (Concat) [19]	Empty model answer
PointLLM (Concat) [21]	No, the first object does not completely satisfy all of these requirements compared to the second object.
ShapeLLM (Concat) [17]	Yes, the first object meets all of the requirements compared to the second object.
Multi-3DLLM (Ours)	**No. The back rest is not narrower as required; instead, it has a curved top.**

Figure 19. **Qualitative Comparison on Change Captioning (Verify Task).**

Example 9: Change Captioning (Delta Caption)



Question: List the geometric adjustments required to turn the second object into the first object.
Ground Truth: Flatten the top surface to create a simple, open block shape.

Model	Response
LLaVA (1-view) [14]	Modify the shape to make it thicker and wider, with a more rounded edge.
Molmo (1-view) [8]	To transform the second object into the first object, we would need to make the following geometric adjustments: 1. Extend the overall length of the object by approximately 20-30%. 2. Increase the width by about 10-15%. 3. Raise the height by roughly 15-20%. 4. Smooth out the jagged edges, particularly on the right side, to create a more uniform rectangular shape. 5. Add a small protrusion or tab on the right side, similar to the one in the first object. 6. Refine the top edge to create a more defined flat surface. 7. Adjust the bottom edge to ensure it's straight and even with the new dimensions. These changes would transform the second object's more abstract, rectangular-like shape into the first object's more defined and precise rectangular form.
MiniGPT-3D (Concat) [19]	the model is a 3d object, so it can be rotated, translated, or viewed from different angles.
PointLLM (Concat) [21]	The 3D model represents a simple, minimalistic stool. It has a flat surface on top, supported by three legs. The legs are cylindrical and equally spaced. The stool lacks any additional features or decorations. Given the model, what text prompt would you provide to modify it into a chair?
ShapeLLM (Concat) [17]	Modify the second shape to resemble the first by making it thicker, shorter, and more curved, with a narrower seat and a taller backrest.
Multi-3DLLM (Ours)	**Transform the shape to be rectangular, add a flat top, and ensure it has a smooth surface without any protrusions.**

Figure 20. **Qualitative Comparison on Change Captioning (Delta Captioning Task).**

References

[1] Panos Achlioptas, Ian Huang, Minhyuk Sung, Sergey Tulyakov, and Leonidas Guibas. ShapeTalk: A language dataset and framework for 3d shape edits and deformations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12685–12694, 2023. 4

[2] Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. Qwen technical report. *arXiv preprint arXiv:2309.16609*, 2023. 1

[3] Satanjeev Banerjee and Alon Lavie. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72, 2005. 8

[4] Sijin Chen, Xin Chen, Chi Zhang, Mingsheng Li, Gang Yu, Hao Fei, Hongyuan Zhu, Jiayuan Fan, and Tao Chen. Ll3da: Visual interactive instruction tuning for omni-3d understanding reasoning and planning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 26428–26438, 2024. 5

[5] Yun-Chun Chen, Haoda Li, Dylan Turpin, Alec Jacobson, and Animesh Garg. Neural shape mating: Self-supervised object assembly with adversarial shape priors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, page 12724–12733, 2022. 3

[6] Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E Gonzalez, Ion Stoica, and Eric P Xing. Vicuna: An open-source chatbot impressing GPT-4 with 90%* ChatGPT quality. lmsys.org, 2023. 1

[7] Matt Deitke, Dustin Schwenk, Jordi Salvador, Luca Weihs, Oscar Michel, Eli VanderBilt, Ludwig Schmidt, Kiana Ehsani, Aniruddha Kembhavi, and Ali Farhadi. Objaverse: A universe of annotated 3d objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13142–13153, 2023. 1

[8] Matt Deitke, Christopher Clark, Sangho Lee, Rohun Tripathi, Yue Yang, Jae Sung Park, Mohammadreza Salehi, Niklas Muennighoff, Kyle Lo, Luca Soldaini, Jiasen Lu, Taira Anderson, Erin Bransom, Kiana Ehsani, Huong Ngo, YenSung Chen, Ajay Patel, Mark Yatskar, Chris Callison-Burch, Andrew Head, Rose Hendrix, Favyen Bastani, Eli VanderBilt, Nathan Lambert, Yvonne Chou, Arnavi Chheda, Jenna Sparks, Sam Skjonsberg, Michael Schmitz, Aaron Sarnat, Byron Bischoff, Pete Walsh, Chris Newell, Piper Wolters, Tanmay Gupta, Kuo-Hao Zeng, Jon Borchardt, Dirk Groeneveld, Crystal Nam, Sophie Lebrecht, Caitlin Wittlif, Carissa Schoenick, Oscar Michel, Ranjay Krishna, Luca Weihs, Noah A. Smith, Hannaneh Hajishirzi, Ross Girshick, Ali Farhadi, and Aniruddha Kembhavi. Molmo and pixmo: Open weights and open data for state-of-the-art vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 91–104, 2025. 16, 17, 18, 19, 20, 21, 22, 23, 24

[9] Kavita Ganesan. Rouge 2.0: Updated and improved measures for evaluation of summarization tasks. *arXiv preprint arXiv:1803.01937*, 2018. 8

[10] Tianyu Gao, Xingcheng Yao, and Danqi Chen. Simcse: Simple contrastive learning of sentence embeddings. *arXiv preprint arXiv:2104.08821*, 2021. 8

[11] Haifeng Huang, Yilun Chen, Zehan Wang, Rongjie Huang, Runsen Xu, Tai Wang, Luping Liu, Xize Cheng, Yang Zhao, Jiangmiao Pang, et al. Chat-scene: Bridging 3d scene and large language models with object identifiers. *Advances in Neural Information Processing Systems*, 37: 113991–114017, 2024. 5

[12] Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547, 2019. 1

[13] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81, 2004. 8

[14] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916, 2023. 16, 17, 18, 19, 20, 21, 22, 23, 24

[15] Tiange Luo, Chris Rockwell, Honglak Lee, and Justin Johnson. Scalable 3d captioning with pretrained models. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 75307–75337, 2023. 1

[16] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318, 2002. 8

[17] Zekun Qi, Runpei Dong, Shaochen Zhang, Haoran Geng, Chunrui Han, Zheng Ge, Li Yi, and Kaisheng Ma. ShapeLLM: Universal 3d object understanding for embodied interaction. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 214–238, 2024. 16, 17, 18, 19, 20, 21, 22, 23, 24

[18] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 8748–8763. PmLR, 2021. 1, 6

[19] Yuan Tang, Xu Han, Xianzhi Li, Qiao Yu, Yixue Hao, Long Hu, and Min Chen. Minigpt-3d: Efficiently aligning 3d point clouds with large language models using 2d priors. In *Proceedings of the ACM International Conference on Multimedia (ACM MM)*, page 6617–6626, 2024. 16, 17, 18, 19, 20, 21, 22, 23, 24

[20] Yuan Tang, Xu Han, Xianzhi Li, Qiao Yu, Jinfeng Xu, Yixue Hao, Long Hu, and Min Chen. More text, less point: Towards 3d data-efficient point-language understanding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 7284–7292, 2025. 6

[21] Runsen Xu, Xiaolong Wang, Tai Wang, Yilun Chen, Jiangmiao Pang, and Dahua Lin. Pointllm: Empowering large

language models to understand point clouds. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 131–147, 2024. 1, 6, 7, 16, 17, 18, 19, 20, 21, 22, 23, 24

[22] Le Xue, Ning Yu, Junnan Li, Roberto Martín-Martín, Jiajun Wu, Ran Xu, Juan Carlos Niebles, Caiming Xiong, and Silvio Savarese. ULIP-2: Towards scalable multimodal pre-training for 3D understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 27091–27101, 2024. 1

[23] Xumin Yu, Lulu Tang, Yongming Rao, Tiejun Huang, Jie Zhou, and Jiwen Lu. Point-BERT: Pre-training 3D point cloud transformers with masked point modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19313–19322, 2022. 1, 3

[24] Qingnan Zhou and Alec Jacobson. Thingi10k: A dataset of 10,000 3d-printing models. *arXiv preprint arXiv:1605.04797*, 2016. 3